Secure and Safe Control of Connected and Automated Vehicles Against False Data Injection Attacks

Guoxi Chen, Student Member, IEEE, Tiejun Wu, Xinde Li, Senior Member, IEEE, and Ya Zhang, Senior Member, IEEE

Abstract—This paper studies the secure and safe control problem of connected and automated vehicles (CAVs) with false data injection (FDI) attacks. A secure and safe controller with a novel surrounding vehicles’ state estimator and an attack detector is proposed. The state estimation for surrounding vehicles is collectively processed by combining the deep neural network-based predictions with model-based estimations. Additionally, a weight in the loss function is proposed for more accurate predictions of vehicles that are closer to the ego vehicle. For attack detection, a novel scheme that utilizes control outcomes to train detection actions is proposed. Different from the objectives of existing detectors, the reward for the proposed detector is designed to encourage the CAV to fully utilize the observations. A reward setting and the decision preference of the ego vehicle are theoretically analyzed. The effectiveness of the proposed algorithm is validated in an open simulation environment.

Index Terms—Secure control, safe control, connected and automated vehicles, false data injection attacks, artificial intelligence.

I. INTRODUCTION

The development of connected and automated vehicles (CAVs) is revolutionizing the transportation systems. Through wireless communication and sensing technologies, CAVs can share their local measurements with surrounding vehicles, which enhances safety, efficiency, and overall throughput [1]. However, vehicular networks are susceptible to threats from malicious attackers, making the observation used for decision-making unreliable [2].

Malicious attackers can be classified into active and passive. Passive attackers such as eavesdroppers, aim to obtain private information from the vehicles, while active attackers compromise the observations of the CAVs, and threaten the vehicles’ decision-making process [3], [4], [5], [6]. Active attacks can generally be divided into denial-of-service (DoS) attacks and false data injection (FDI) attacks. DoS attacks happen when an adversary floods the communication channel with redundant signals, thereby obstructing information flow and delaying its reach to the intended destination [7], [8]. FDI attacks deceive CAVs by sending false observations to the controller [9], [10], and/or interfere with the normal execution of the controller [11], [12]. In this paper, we are dedicated to developing a secure and safe control strategy to adaptively utilize the observations in environments with FDI attacks. To achieve this goal, the process from observing to making a controller decision can be divided into four steps: estimating current observations based on historical information; judging the existence of FDI attacks from the relationship between the estimation and the observation; choosing the estimation or the observation based on the detection results; and ultimately determining the ego vehicle’s control action based on the result of the previous step.

There have been many research on the model-dependent trajectory estimation. In [12] and [13], the model parameters of the generalized system are assumed to be known. In specific studies on CAVs, the corresponding physical models are proposed, and the relevant physical parameters are often assumed to be known [9], [14], [15], [16], [17]. Once the model of the system can be determined, a series of methods such as Kalman filter and its nonlinear extension can be used [9], [17]. Maximum likelihood estimation is also deployed to fuse real-time and historical probe vehicle trajectory [18]. The physical model always assumes that the vehicle maintains a constant turning rate and velocity [19]. However, the behavior of vehicles is often socially influenced, and their maneuvers can change rapidly within a short period of time, particularly during lane changes. These dynamic aspects, driven by interactions with other traffic participants, cannot be adequately captured by purely physics-based models.

With the advancement of artificial intelligence, deep learning has emerged as an effective solution for estimating vehicle maneuvers [20]. Liu et al. [19] combined physical model and deep conditional generative network to determine vehicle trajectories by predicting lane crossing and final points. Although this scheme takes into account the driver’s driving style, it overlooks the interaction between the vehicle being predicted.
and other surrounding vehicles. Furthermore, while curve fitting eliminates cumulative errors, it compromises short-term estimation accuracy, which is unsuitable for the secure and safe control. To capture interactions, Alahi et al. [21] proposed the social pooling of hidden states, and implemented it in the long short-term memory (LSTM) networks. This scheme has achieved better results in human trajectory prediction. To integrate the interaction module into vehicle trajectory prediction, Deo et al. [22] proposed a parameter-sharing LSTM for each vehicle involved in the prediction, and converted the coordinate information into a spatial grid, known as the social tensor, aiming to explicitly capture the spatial characteristics of the surrounding vehicles. Sheng et al. [23] proposed a graph-based spatial-temporal convolutional network, which employs a graph to describe the relationships between vehicles and utilizes a weighted adjacency matrix to represent the intensity of their mutual influence. These approaches leverage a spatial tensor to catch interactions and integrate them into each agent’s unique trajectory estimator through a decoder. However, these frameworks, constrained by parameter sharing, do not adequately differentiate the importance of predicted targets for the ego-vehicle.

In the field of attack detection, most studies consider the detection process as a component of the observer, which is separate from the design of the controller. The underlying principle of these studies involves comparing estimations with actual observations [12], [14], [15], [16]. If the residual exceeds a certain threshold, the observation is disregarded. Such a threshold is derived based on the covariance of the residuals and the expert experience. There are also some intelligent detection algorithms used for attack detection. In [24], a data-driven attack detection and recognition algorithm was proposed according to subspace identification and compressive sensing theories. They accomplished attack identification using convex optimization methods. The process of attack detection was modeled as a partially observable markov decision process (POMDP) in [13], and the deep reinforcement learning (DRL) technique is used for attack detection. It incentivizes actions that lead to accurate judgments by setting appropriate reward. Regardless of attack detection schemes, agents use the estimation instead of the observation when an attack is identified. However, from the control perspective, identifying all attacks may not always present the optimal solution for the controller. If an attacker injects FDI attacks into observations with high-frequency and low-intensity, the strategy of dismissing observation in favor of reliance on estimations may lead to the accumulation of estimation errors. When the accumulated error exceeds the impact caused by the attack signal, it will no longer be a wise choice to give up the observation. Considering this, we propose a novel detection approach that differs from traditional methods by evaluating the use of observation based on control effectiveness rather than only observation.

For the safe control of CAVs, Katrakazas et al. [25] designed a planning-based method considering both reliability and safety, in which the endpoint was first established and then the optimal smooth path was determined. With the development of artificial intelligence, DRL has been used in the control of CAVs. The decision-making process of a vehicle can be modeled as a Markov decision process (MDP), and DRL is considered to be an effective model-free control solution [26]. Zhang et al. [27] proposed a novel risk-sensitive learning-based algorithm that integrates the risk-identification method and the Lyapunov function within the soft actor critic (SAC) framework. He et al. [28] proposed a multi-objective actor-critic framework to trade off energy consumption and travel efficiency. Wang et al. [29] used DRL to fuse information from multiple sensors to solve the vehicle lane change problem. Chen et al. [30] proposed a novel external spatial attention (ESA) module, which was utilized in the deep Q network (DQN) algorithm to form the ESA-DQN, and designed a lightweight safety layer based on the support vector machine (SVM) to further improve the safety rate.

This paper studies the secure and safe control problem of CAVs with FDI attacks. There are following challenges posed by existing technologies.

1) Existing prediction algorithms [21], [22], [23], which assign an estimator to each vehicle, become ineffective when new targets appear without historical observational trajectories.

2) When equipping each target with an estimator, parameter sharing has become an indispensable technique to reduce parameter size. However, it prevents the estimator from assigning prediction priorities for the ego vehicle.

3) Most of existing attack detection algorithms [12], [13], [14], [15], [16] are judged based on the relationship between observations and predictions. A commonly used approach to dealing with attacks is to discard attacked data, while it would significantly influence the control performance.

In view of the above challenges, this paper proposes a secure and safe control of CAVs against FDI attacks. The main contributions are summarized as follows.

1) A hybrid prediction algorithm combining neural networks and physical models is proposed. It identifies interactions and provides estimation by collectively considering all prediction targets that are centered around the ego vehicle. It intuitively provides a solution for new target prediction that lacks historical trajectories and also enables the prioritization of predictive tasks for the controller.

2) A novel detection approach is proposed. Differing from most of existing FDI attack detectors [8], [12], [13], it aim to adaptively utilize the observations and to minimize the impact of FDI attacks on the safe control. Furthermore, a DRL solution is proposed and a corresponding neural network is designed.

3) The limitation brought by perceptual aliasing is analyzed in the detection process, and the two reward design schemes with different objectives is proposed theoretically.

The rest of this paper is organized as follows. In Section II, the system model is established, and the purpose of the control algorithm is clarified. Section III proposes the estimation algorithm based on deep neural network and physical process. Section IV proposes the secure and safe controller. The effect of different reward on the results is further analyzed. Section V
provides several experimental results in an open simulation environment, and Section VI is a conclusion.

Notation: \( \mathbb{R}^n \) and \( \mathbb{R}^{n \times n_2 \times n_3} \) denote the \( n \)-dimensional Euclidean space and the set of all \( n_1 \times n_2 \times n_3 \) real matrices, respectively. \( \| \cdot \|_F \) is the Frobenius norm of a matrix, and \( \odot \) denotes the Hadamard product, which is the element-wise multiplication of matrices. \( \mathbb{E}[\cdot] \) describes the mathematical expectation. \( \lfloor \cdot \rfloor \) is ceil function, and \( \lceil \cdot \rceil \) gives the smallest integer greater than or equal to \( \cdot \). \( P(A) \) stands for the occurring probability of event \( A \). \( P(A|B) \) is the conditional probability, which represents the probability of \( A \) when \( B \) has occurred. \( \text{clip}(\cdot, z_{\text{low}}, z_{\text{up}}) \) is the clipping function to limit \( \cdot \) in an interval \( [z_{\text{low}}, z_{\text{up}}] \). \( |\cdot| \) means the absolute value.

II. PROBLEM FORMULATION

In this paper, all the vehicles are considered to have the same direction of travel, and the secure and safe control problem of the CAV is studied. The discrete dynamics of ego vehicle can be described as

\[
\begin{align*}
\frac{d}{dt} x^{e}(k+1) &= f(x^{e}(k), \delta(k), a_{vc}(k)), \\
x^{e}(k) &= [x^{e}(k), v^{e}(k), \psi^{e}(k), \beta^{e}(k)]^T
\end{align*}
\]

where \( x^{e}(k) = [x^{e}(k), y^{e}(k), v^{e}(k), \psi^{e}(k), \beta^{e}(k)]^T \) represents the state of the ego vehicle at time \( k \), including the longitudinal and lateral positions, speed, heading angle, and sideslip angle, respectively. The control signals \( a_{vc}(k) \) and \( \delta(k) \) at time \( k \) correspond to acceleration and steering angle, respectively. \( f(\cdot) \) represents a nonlinear function mapping. In model-based studies, the most general is kinematic bicycle model [31]. This model situates the center of gravity at the centroid, neglecting the impacts of intricate details such as mass distribution, tire forces, suspension properties, and aerodynamic effects. This oversimplification may not provide a realistic representation of actual scenarios. In this paper, the process model of the ego vehicle is assumed to be known, and the general nonlinear notation \( f(\cdot) \) is used. The ego vehicle is assumed to be able to obtain its own longitudinal position \( x^{e}(k) \), lateral position \( y^{e}(k) \), speed \( v^{e}(k) \) and heading angle \( \psi^{e}(k) \) through the on-board sensor [32]. In other words, \( x^{e}(k) = [x^{e}(k), y^{e}(k), v^{e}_{x}(k), v^{e}_{y}(k)]^T \) containing the longitudinal position, lateral position, longitudinal speed and lateral speed can be obtained by the controller.

In order to make a secure and safe decision to achieve higher driving efficiency while avoiding collisions, CAVs collect the surrounding vehicles’ state including longitudinal position, lateral position, longitudinal speed and lateral speed by communication [14]. However, the state information transmitted over wireless networks may be compromised by malicious FDI attackers [12]. If compromised observations are used, a controller that would typically operate correctly may display abnormal behavior, posing potentially fatal consequences [15]. The state \( x^{e} \) of the ego vehicle is collected locally, hence it is considered to be secure. However, the state \( x^{e}_{i}(k) = [x^{e}_{x}(k), y^{e}_{i}(k), v^{e}_{x}(k), v^{e}_{y}(k)]^T \) of surrounding vehicle \( i \) collected via the wireless network is vulnerable to attacks. Since attackers are energy constrained, they cannot continue the attack for a long time and often inject attacks intermittently [9], [10], [11], [12], [13], [33]. Thus, at time \( k \), the state of the \( i \)-th surrounding vehicle received by the ego vehicle \( \bar{x}^{e}_{i}(k) \) is

\[
\bar{x}^{e}_{i}(k) = x^{e}_{i}(k) + \alpha_{i}(k)q_{i}(k),
\]

where \( x^{e}_{i}(k) \) is the true state of the \( i \)-th surrounding vehicle at time \( k \). \( \alpha_{i}(k) \in [0, 1] \) indicates whether there is an attack on the state of the surrounding vehicle \( i \) at time \( k \). If there is an attack on the observation of vehicle \( i \) at time \( k \), \( \alpha_{i}(k) = 1 \), otherwise \( \alpha_{i}(k) = 0 \). \( q_{i}(k) \) represents the injected malicious data at time \( k \). This model represents the generic FDI attacks [34]. The following assumptions about the policies of surrounding vehicles and the attackers are proposed in this scenario.

Assumption 1: The control policies of surrounding vehicles are time-invariant.

Assumption 2: The attacker follows an unknown time-invariant attack strategy.

The time-invariant strategy implies that the distribution of actions for a specific scenario remains consistent. Assumption 1 guarantees both the predictability of vehicle maneuver and a stationary environment. In practical scenarios, vehicle strategies are influenced by the driver’s cognitive processes and psychological factors. As a driver gains more experience, his/her driving strategy naturally evolves and improves over time. However, in the short term, the driving strategy remains mostly unchanged. Thus, it is reasonable to assume that the driving strategy remains consistent [27], [30], [35]. Assumption 2 restricts the detection algorithm to a single type of attack, which is a common assumption [12], [13], [15], [16]. The use of deep neural networks makes the algorithm generalizable and may be applicable to other similar types of attack detection.

The ego vehicle uses an \( N + 1 \)-length window to save the observed history of the surrounding vehicles and the ego vehicle, which is expressed as \( \bar{H}(k) = \{x^{e}(k) − N, \{x^{e}_{i}(k) − N\}_{i=1}^{n(k−N)}, \ldots, x^{e}(k), \{\bar{x}^{e}_{i}(k)\}_{i=1}^{n(k)}\} \), where \( \{\bar{x}^{e}_{i}(k) − N\}_{i=1}^{n(k−N)} \) is the succinct representation of set \( \{\bar{x}^{e}_{i}(k) − N | 1 \leq i \leq n(k−N)\} \), and \( n(k) \) denotes the number of surrounding vehicles that can be observed at time \( k \). This paper focuses on obtaining optimal secure and safe control actions to achieve higher driving efficiency while ensuring collision avoidance under Assumption 1 and 2 by using the historical state \( \bar{H}(k) \) and current observation \( \{x^{e}(k+1), \{\bar{x}^{e}_{i}(k+1)\}_{i=1}^{n(k+1)}\} \) at time \( k + 1 \).

III. ESTIMATION ALGORITHM DESIGN

The framework of the secure and safe controller proposed by this paper involves state estimation of surrounding vehicles, safe controller design, and an FDI attack detector to evaluate whether the control action derived from observations or estimations align more closely with those obtained from ground truth. In this section, a state estimation algorithm is firstly designed.

The estimation algorithm provides an estimation of the states of the surrounding vehicles. When designing it, we assume that the uncompromised training data can be collected and trained. Since the state \( x^{e} \) is secure, the main
objective of the estimator network is to estimate the state of surrounding vehicles at time \( k \) based on the historical trajectories. In this section, we first provide a notation list in Table I to collect important and complex symbols to improve readability, and then introduce the model in detail.

Generally, the discrete dynamics of the transmitted information from surrounding vehicle \( i \) can be modeled as

\[
x_i^t(k+1) = g(x_i^t(k), k) + \Delta_i(k+1),
\]

where \( g(\cdot) \) represents the dynamics that can be modeled by physical process, which is known to the ego vehicle. \( \Delta_i(k+1) \) contains many items, and its key factor is the decision made by the surrounding vehicle \( i \) from time \( k \) to \( k+1 \) on the state at time \( k+1 \). Thus, \( \Delta_i(k+1) \) is treated as the maneuver of the surrounding vehicle \( i \) and is unknown for the ego vehicle. This model exclusively represents the dynamics of physical quantities transmitted to the ego vehicle. Our focus lies on the dynamics of physical quantities that vehicle \( i \) transmits to the ego vehicle.

We use a deep neural network to estimate the maneuvers. All the surrounding vehicles are estimated as a whole and represented by an occupancy grid, which helps manage the varied input sizes at different times. The physical model (3) is considered, and the neural network fits the increment between the location estimated by the physical process and the actual location. This increment represents the maneuver of all the surrounding vehicles denoted by an occupancy grid.

The specific details of the proposed algorithm is shown in Fig. 2, which consists of a convolutional LSTM (ConvLSTM) [36] encoder, squeeze and excitation block (SE block) [37] and a maneuver decoder with physics process.

The encoder includes the information gridization and the ConvLSTM-based feature extraction. Information gridization aims to map the trajectory described by raw features into a trajectory described by the occupancy grid. An information gridization includes \( N+1 \) state gridizations. Each state gridization is designed to transform raw features at a single moment, which is represented as \( \hat{X}(k) \), into an occupancy grid with dimensions of \( C \times H \times W \), where \( C \), \( H \), and \( W \) represent the number of grids associated with features, lanes, and driving visibility, respectively. Further details of the conversion process is provided as follows.

In this paper, the dimension of the vehicle feature space \( C \) is equal to 4. For the ego vehicle, the raw features include position and speed as shown in Table I, and the feature vector filled in the occupancy grid is \( \frac{F_{x_i}(k)}{N} \). As for surrounding vehicle \( i \), relative position and velocity are used as raw features, and the feature vector that is ultimately used is \( \frac{F_{x_i}(k)}{N} \), \( W \) represents the number of lanes involved in the scenario, and \( H = \frac{P}{T} \). Then, the \( t \)-th element in the second dimension stores the features within the longitudinal distance interval \([t-1] \times l \), \( t \times l \) from the selected origin, which is related to each instance. For each instance within \( T \), longitudinal position of the ego vehicle at time \( k - N \), i.e., \( x^t(k - N) \), is defined as the origin. Thus, the expression for \( D \) in Table I ensures that all instance within \( T \) can be mapped from the raw features to the corresponding grid. The feature vectors of the ego vehicle and surrounding vehicles are inserted into this grid based on their locations. If there is no raw features corresponding to a particular position, it is filled with zero.

As defined in Table I, we denote the state gridization from the raw features to the occupancy grid as \( T(\chi, I) \). This mapping function requires two inputs: the selected longitudinal origin \( \chi \), and the raw features set \( I \) that needs to be transformed. It returns an occupancy grid with dimensions of \( C \times H \times W \).

For each instance, information gridization involves transforming the input \( H(k) \) into \( H_{x^t(k-N)}(k) \). Then, the label \( Y(k + 1) \) is also transformed through state gridization into \( Y_{x^t(k-N)}(k+1) \). The label is adjusted to include \( x^t(k+1) \) to align with the estimation and account for the influence of the ego-vehicle on the maneuvers of surrounding vehicles. Note that after information gridization, an input with image-like properties is obtained. To extract spatial information from sequences, ConvLSTM encoder is used.

After the encoder maps the features from the original 4 channels to a higher dimension, the SE block [37] is employed to dynamically recalibrate the feature responses for each channel, effectively capturing the interdependencies among them.

The decoder includes a maneuver predictor based on the neural network and a state estimator based on the physical process. The maneuver predictor employs two fully connected layers (FCs). As the neural network acquires only the maneuver of each vehicle, it needs to be supplemented with the state estimation based on physical processes to achieve the final prediction. The final estimate provided by the decoder is presented in an occupancy grid.

The loss function of the estimator network is

\[
L = \sum_{b_1} \| W \circ (\hat{Y}_{x^t(k-N)}(k+1) - Y_{x^t(k-N)}(k+1)) \|^2 \tag{4}
\]

where \( \hat{Y}_{x^t(k-N)}(k+1) = E_{\theta_1}(H(k)) \) represents the output of the decoder when the input is \( H(k) \). \( W \in \mathbb{R}^{C \times H \times W} \) is the weight matrix, which assigns varying degrees of focus to the estimation accuracy at different positions. In the secure and safe control problem, \( W \) can effectively differentiate the importance of surrounding vehicle estimations in different regions. Various techniques including input normalization, momentum update, and early stopping are employed to optimize the performance. Fig. 1 shows the flow chart from the historical trajectory \( H(k) \) to the estimation \( \hat{Y}_{x^t(k-N)}(k+1) \). Since we integrate all the vehicles for prediction, the estimation is a tensor.

The trained estimator network is denoted as \( E_{\theta_1}(\cdot) \), which is a map from a set of historical tracking data to \( \mathbb{R}^{C \times H \times W} \). Since the origin is recorded, it can be restored to its original features including position and speed as the final output \( \hat{Y}(k+1) \) of the model. We simply denote the inverse transformation as \( T^{-1} \), which is a map from \( \mathbb{R}^{C \times H \times W} \) to a set.

**Remark 1**: In comparison to prediction-focused studies, the proposed estimation algorithm does not apply the decoder that converts the occupancy grid (known as the social tensor) into individual vehicle trajectories, but directly integrates it with physical models for prediction. As shown in (4), we directly
TABLE I

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>The uncompromised training set</td>
<td></td>
</tr>
<tr>
<td>( N )</td>
<td>Feature scaling factor</td>
<td></td>
</tr>
<tr>
<td>( I )</td>
<td>The length of each grid</td>
<td></td>
</tr>
<tr>
<td>( \hat{x}_s^i(k+1) )</td>
<td>Estimation of raw features of vehicles ( i ) at time ( k+1 )</td>
<td></td>
</tr>
<tr>
<td>( \hat{T}(X,I) )</td>
<td>The estimator’s mapping function from raw features to occupancy grids</td>
<td></td>
</tr>
<tr>
<td>( \hat{E}_b(\cdot) )</td>
<td>The estimator neural network with parameters ( \theta_1 )</td>
<td></td>
</tr>
<tr>
<td>( \hat{b}_y )</td>
<td>Batch size set for estimator training</td>
<td></td>
</tr>
<tr>
<td>( \hat{H}(k) )</td>
<td>The track history denoted by raw features</td>
<td>( {X(k-N),...,X(k-1),X(k)} )</td>
</tr>
<tr>
<td>( Y(k+1) )</td>
<td>Ground truth of raw features of surrounding vehicles at time ( k+1 )</td>
<td>( {x^s_i(k+1)}^N_{i=1} )</td>
</tr>
<tr>
<td>( \hat{Y}(k+1) )</td>
<td>Estimation of raw features of surrounding vehicles at time ( k+1 )</td>
<td>( {\hat{x}^s_i(k+1)}^N_{i=1} )</td>
</tr>
<tr>
<td>( \mathcal{K} )</td>
<td>All the time instances included in the ( T )</td>
<td>( k \in \mathcal{K} ) if ( {\hat{H}(k),X(k+1)} \in T )</td>
</tr>
<tr>
<td>( D )</td>
<td>The total distance</td>
<td>( \max_{k \in \mathcal{K}} {\hat{x}^s_i(k+1) - x^s_i(k-N)} )</td>
</tr>
<tr>
<td>( F_s(k) )</td>
<td>Raw features of ego vehicle</td>
<td>( [0,y_1(k),y_2(k),y_3(k)]^T )</td>
</tr>
<tr>
<td>( F^s(k) )</td>
<td>Raw features of surrounding vehicles</td>
<td>( [\hat{x}^s_i(k+1)-x^s_i(k)] )</td>
</tr>
<tr>
<td>( \hat{X}(k) )</td>
<td>Uncompromised position and speed of all the vehicles at time ( k )</td>
<td>( {\hat{x}_s^i(k),\hat{y}_s^i(k),\hat{v}_s^i(k)} )</td>
</tr>
<tr>
<td>( X_{\hat{X}^s(k-N)}(k) )</td>
<td>Occupancy grid representation of ( X(k) ) with origin at ( x^s(k-N) )</td>
<td>( \mathbb{T}[x^s(k-N),(X(k))] )</td>
</tr>
<tr>
<td>( H_{\hat{X}^s(k-N)}(k) )</td>
<td>Track history denoted by the occupancy grid</td>
<td>( {X_{\hat{X}^s(k-N)}(k-N),...,X_{\hat{X}^s(k-N)}(k-1),X_{\hat{X}^s(k-N)}(k)} )</td>
</tr>
<tr>
<td>( Y_{\hat{X}^s(k-N)}(k+1) )</td>
<td>Occupancy grid representation of ( Y(k+1) ) with origin at ( x^s(k-N) )</td>
<td>( \mathbb{T}[x^s(k-N),(x^s(k+1),Y(k+1))] )</td>
</tr>
<tr>
<td>( Y_{\hat{X}^s(k-N)}(k+1) )</td>
<td>Occupancy grid representation of ( Y(k+1) ) with origin at ( x^s(k-N) )</td>
<td>( Y_{\hat{X}^s(k-N)}(k+1) = \hat{E}_b(\hat{h}(k)) )</td>
</tr>
</tbody>
</table>

Remark 2: We use an encoder-decoder framework with appropriate input and output formats for maneuver prediction. However, the secure and safe control approach proposed in this paper is not limited to any specific neural network structure, which means it can be substituted with any framework that has suitable input and output formats and performs well.

IV. CONTROL ALGORITHM DESIGN

In this section, a secure and safe control algorithm is proposed, which is divided into two subsections. In the first subsection, we provide a safe controller for a safe and reliable control scheme within an attack-free environment. Then, the secure control algorithm is proposed to provide secure control in the presence of FDI attacks.

A. Safe Control Algorithm Design

The control issue for CAVs can be framed as an MDP, with the state comprising both the position and velocity of the ego vehicle and surrounding vehicles. Based on Assumption 1, the state transition probability is stationary. Thus, a DRL controller can be utilized in this scenario. In order to increase readability, we present the important and potentially confusing notations in this subsection in Table II.

In order to maintain permutation invariance [30], we employ a gridization technology for controller, which is similar to state gridization in Section III. There are notable distinctions in three crucial aspects: the longitudinal origin, the overall grid distance, and the number of lanes. In this section, when transforming \( X(k) \) at time step \( k \), the longitudinal origin is chosen as the longitudinal position \( x^s_i(k) \) of the ego vehicle. Furthermore, the total distance is the limit of the view distance of the ego vehicle at every time step. Considering that control is centered around the ego vehicle, the lane information should at most include only the current lane of the ego vehicle and its adjacent lanes on either side. Unlike the transformation results in Section III, this section involves converting raw features into an occupancy grid with dimension \( C \times H' \times W' \) for controller input, where \( W' = \min\{3,W\} \) and \( H' = \lceil \frac{D}{z} \rceil \).

 TABLE II

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D' )</td>
<td>The view distance of the ego vehicle</td>
</tr>
<tr>
<td>( \hat{T}(X,I) )</td>
<td>The mapping function of the controller</td>
</tr>
<tr>
<td>( a_x(k) )</td>
<td>Action of the DRL controller at time ( k )</td>
</tr>
<tr>
<td>( a_{cc}(k) )</td>
<td>Acceleration signal of the ego vehicle</td>
</tr>
<tr>
<td>( v^s(k) )</td>
<td>Speed of the ego vehicle at time ( k )</td>
</tr>
<tr>
<td>( v_i(k) )</td>
<td>Target speed of the ego vehicle at time ( k )</td>
</tr>
<tr>
<td>( \delta(k) )</td>
<td>Steering angle signal of the ego vehicle at time ( k )</td>
</tr>
<tr>
<td>( z )</td>
<td>Lane width</td>
</tr>
<tr>
<td>( \Delta_{abs}(k) )</td>
<td>Lateral position of the lane center line at time ( k )</td>
</tr>
<tr>
<td>( \gamma' )</td>
<td>Discount factor set during controller training</td>
</tr>
<tr>
<td>( Q(s,a;\theta_2) )</td>
<td>Deep Q network of controller with parameters ( \theta_2 )</td>
</tr>
<tr>
<td>( b_{r_2} )</td>
<td>Batch size set for controller training</td>
</tr>
</tbody>
</table>
The DRL controller action \( a_c(k) \) is in the action space \([0, 1, 2, 3, 4]\). These five high-level discrete actions correspond to lane change left, keep state, lane change right, accelerate and decelerate, respectively. They are mapped to the underlying control signal \( a_{cc}(k) \) and \( \delta(k) \) via a proportional controller. For the longitudinal controller, the mapping law is

\[
a_{cc}(k) = K_p(v_r(k) - v^e(k)),
\]

where \( K_p \) is a preset controller parameter. Acceleration (action 3) and deceleration (action 4) are mapped to increase and decrease of \( v_r(k) \), respectively. For steering controller, the mapping law is

\[
\delta(k) = \arcsin \left( \frac{z w_r(k)}{2 v^e(k)} \right),
\]

where \( w_r(k) = K_{\psi}(\psi^*(k) - \psi(k)) \) and \( \psi^*(k) = \arcsin \left( \frac{K_{\psi}(\Delta_{\text{lat}}(k))}{v^e(k)} \right) \), where \( K_{\text{lat}} \) and \( K_{\psi} \) are the preset controller parameters. Lane change left (action 0) and lane change right (action 2) correspond to the increase and decrease of \( \Delta_{\text{lat}}(k) \), respectively.

Since we encourage efficient driving and safe actions, the reward of the driving strategy is chosen as

\[
r^e(k) = \begin{cases} 0, & \text{Out of lane}, \\ \lambda * r^c_1(k) + r^c_2(k) + 1, & \text{Others}, \end{cases}
\]

where \( \lambda \) is the preset weighting factor for \( r^c_1(k) \), \( r^c_1(k) \) represents the reward for efficient driving with following expression

\[
r^c_1(k) = \text{clip}(\frac{v^c_r(k) - v_{\min}}{v_{\max} - v_{\min}}, 0, 1),
\]

where \( v_{\max} \) and \( v_{\min} \) represent the maximum and minimum speed distinguished in the reward value setting, respectively. \( r^c_2(k) \) is the penalty caused by the collision, which is designed as follows

\[
r^c_2(k) = \begin{cases} -1, & \text{Collision}, \\ 0, & \text{Others}. \end{cases}
\]

An ESA-DQN [30] is employed to solve the control problem, and the following general loss function is adopted.

\[
L(\theta_2) = \sum_{b_2} \left( r(s, a_c) + \gamma' \max_{a'} Q(s', a'; \theta_2^\pi) - Q(s, a_c; \theta_2) \right)^2,
\]

where \( Q(s, a; \theta_2^\pi) \) is the target network, which is periodically updated from the latest \( Q(s, a; \theta_2) \) to ensure stability.

To further improve safety, a lightweight safety layer based on SVM is employed, which is trained using dangerous actions that have led to collisions during the training process. The controller used in this subsection is the same as [30]. The final trained controller network is denoted as \( Q^e_2 : \mathbb{R}^C \times H \times W \rightarrow \mathbb{R}^5 \) with parameters \( \theta_2^e \).

Remark 3: Since DRL only provides a soft constraint against collisions, the controller is augmented with an additional safety layer composed of an SVM, which essentially functions to identify the boundary of action safety. As a linear classifier, it offers significant interpretability. Its trained parameters can be interpreted as some rule-based hard constraints.

The training process of the estimation and control algorithm without FDI attacks is summarized in Algorithm 1.

B. Secure Control Algorithm Design

In this subsection, a secure control algorithm is designed for ego vehicle in the presence of FDI attackers. During the training of the detector, the parameters \( \theta_1^e \) and \( \theta_2^e \) are fixed, which are trained by the uncompromised training data. Unlike existing studies, the detector is designed to minimize the impact of FDI attacks. The attack detection process is modeled as a POMDP, and the reward is designed to encourage detection actions that align with the control actions when there are no attacks. The detailed analysis is divided into modeling, feasibility analysis and DRL solution. In order to increase readability, we present the important and potentially confusing notations in this subsection in Table III.
Algorithm 1 Training Process of the Estimation and Control Algorithm for Ego Vehicle Without FDI Attacks

1: **output:** $\theta_1^2$, $\theta_2^2$ and trained safety layer.

2: **initialization:** $B$: empty replay buffer; $\mathcal{K}$: training timesteps; $\theta_1, \theta_1^2, \theta_2, \theta_2^2$: initial network parameters; $\mathbb{U}$: update frequency; $k = 0$.

3: while $k < \mathcal{K}$ do:

4: Choose $a_c(k)$ according to $Q_{\theta_2}$ by $\epsilon$-policy;

5: Obtain $r^c(k)$ according to (7);

6: Obtain $X(k + 1)$ from the environment;

7: Add $\{X(k), a_c(k), r(k), X(k + 1)\}$ to $B$;

8: $k \leftarrow k + 1$;

9: if the environment arrives a terminal:

10: reset the environment;

11: end if

12: if $k > b_1$:

13: Sample $b_{r_1}$ tuples from $B$;

14: Update $\theta_1$ according to (10);

15: if $k \% \mathbb{U} == 0$:

16: $\theta_2^2 \leftarrow \theta_2$;

17: end if

18: end if

19: if $k > b_{r_2} + N + 1$:

20: Sample $b_{r_2}$ data from $B$;

21: Update $\theta_2$ data to (4);

22: end if

23: end while

24: Train the safety layer using the data in $B$.

### TABLE III
**NOTATION LIST OF THE DETECTOR**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>Subspace of states encouraged to accept observations</td>
<td>$S^0$</td>
</tr>
<tr>
<td>$S^1$</td>
<td>Subspace of states encouraged to accept estimations</td>
<td>$S^1$</td>
</tr>
<tr>
<td>$\bar{S}$</td>
<td>Subspace of states where perceptual aliasing may occur</td>
<td>$\bar{S}$</td>
</tr>
<tr>
<td>$\bar{S}^|$</td>
<td>Subspace of states where perceptual aliasing does not occur</td>
<td>$\bar{S}^|$</td>
</tr>
<tr>
<td>$b_{r_2}$</td>
<td>Batch size of the detector training</td>
<td>$b_{r_2}$</td>
</tr>
<tr>
<td>$Q^*(\alpha, \theta_2)$</td>
<td>Deep Q network of detector with parameters $\theta_2$</td>
<td>$Q^*(\alpha, \theta_2)$</td>
</tr>
<tr>
<td>$Y(k)$</td>
<td>Observation or estimation recorded in the historical tracks at time $k$</td>
<td>$Y(k)$</td>
</tr>
<tr>
<td>$H(k)$</td>
<td>Historical tracks stored at time $k$</td>
<td>$H(k)$</td>
</tr>
<tr>
<td>$\tilde{X}(k)$</td>
<td>Observation including ego vehicle and surrounding vehicles at time $k$</td>
<td>$\tilde{X}(k)$</td>
</tr>
<tr>
<td>$\tilde{X}(k)$</td>
<td>Estimation including ego vehicle and surrounding vehicles at time $k$</td>
<td>$\tilde{X}(k)$</td>
</tr>
<tr>
<td>$s(k)$</td>
<td>Hidden state of the environment at time $k$</td>
<td>$s(k)$</td>
</tr>
<tr>
<td>$a_c(k)$</td>
<td>Controller action obtained according to $\tilde{X}(k)$ and parameters $\theta_2^*$</td>
<td>$a_c(k)$</td>
</tr>
<tr>
<td>$A(k)$</td>
<td>Detection actions in the $N + 1$-length observation window at time $k$</td>
<td>$A(k)$</td>
</tr>
<tr>
<td>$\xi(k)$</td>
<td>Auxiliary variables at time $k$</td>
<td>$\xi(k)$</td>
</tr>
</tbody>
</table>

1) **POMDP Modeling:** The detection problem is described as a discrete-time POMDP that includes the seven-tuple $(S, A, T, O, R, G, P)$, where $S$, $A$, $T$, $O$, and $R$ represent the state space, action space, observation space and reward space, respectively. $T$ is the set of the unknown conditional transition probabilities between the hidden states, $O$ is the observation model and $P$ is the strategy set.

In this scenario, the hidden state includes observations, estimations, detect actions and attack signals as shown in Table III. The detection agent uses discrete action space $A = \{1, 0\}$, which represents the adoption of estimation and observation, respectively. Mathematically, the action $a(k) \in A$ of the detection agent is

$$a(k) = \begin{cases} 1, & \text{Use estimation,} \\ 0, & \text{Use observation.} \end{cases}$$

We use $\tilde{a}_c(k)$ and $\tilde{a}_c(k)$ to represent the actions that are obtained by the controller network $Q_{\theta_2}$ with the trained safety layer based on $\tilde{X}(k)$ and $\tilde{X}(k)$, respectively. The detection action $a(k)$ ultimately affects the final control action $a_c(k)$ of the ego vehicle, and their relationship is

$$a_c(k) = \begin{cases} \tilde{a}_c(k), & \text{when } a(k) = 0, \\ \tilde{a}_c(k), & \text{when } a(k) = 1. \end{cases}$$

Since the signals of the attacker are unobservable, the agent’s observations $o(k) \in O$ are recorded as $o(k) = \{\tilde{X}(k), \tilde{X}(k), A(k)\}$.

The reward function $r : S \times A \rightarrow R$ at time $k$ represents the reward obtained from the environment.

The ego vehicle sets an $N + 1$ window to save the historical tracks of the surrounding vehicles and itself. Observations at these moments are denoted as $\dot{H}(k)$ in the previous section. However, the data utilized by the estimator at time $k + 1$ is not $\dot{H}(k)$. Some observations may be replaced by estimations since they at that time are considered to be attacked. To avoid any potential confusion in symbols, we use $\dot{H}(k)$ to denote...
the historical tracks as shown in Table III, and
\[
\bar{Y}(k) = \left\{ \begin{array}{ll}
\tilde{Y}(k), & \text{when } a(k) = 1,
\tilde{\hat{Y}}(k)_{i=1}^{n(k)}, & \text{when } a(k) = 0.
\end{array} \right.
\]
(13)

To explain the rationale behind the reward setting, we first analyze the effects of different detection actions on control actions. Choosing different detection actions means choosing different inputs for the controller network, which ultimately leads to different Q-value for the 5 control actions. If the estimation is adopted, Q-value \( \tilde{q}(k+1) \) of the 5 actions is
\[
\tilde{q}(k) = Q_{o_2}^* T'(x^*(k), \tilde{X}(k)).
\]
(14)

If the observation is adopted, Q-value \( \bar{q}(k) \) of the 5 actions is
\[
\bar{q}(k) = Q_{o_2}^* T'(x^*(k), \bar{X}(k)).
\]
(15)
The true q-values \( q(k) \) of these 5 actions at time \( k \) are denoted as
\[
q(k) = Q_{o_2}^* T'(x^*(k), X(k)).
\]
(16)

To facilitate the description of reward, a sorting function \( \sigma \) and a comparison function \( \Phi \) are defined, respectively. For a five-dimensional vector \( q^o \in \mathbb{R}^5 \), a new vector \( q^a = [a_1, a_2, a_3, a_4, a_5]^T \) is defined through \( \sigma : \mathbb{R}^5 \rightarrow \mathbb{R}^5 \), where \( a_i \) represents the position of the \( i \)-th largest element in vector \( q^o \). The function \( \Phi : \mathbb{R}^5 \times \mathbb{R}^5 \rightarrow [0, 1]^5 \) compares the similarity between vectors. Assuming that \( q^a = [a_1, a_2, a_3, a_4, a_5]^T \in \mathbb{R}^5 \) and \( q'^a = [a_1', a_2', a_3', a_4', a_5'^T] \in \mathbb{R}^5 \), their comparison result \( \phi = \Phi(q^a, q'^a) = [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5]^T \) is defined through
\[
\phi_i = \left\{ \begin{array}{ll}
1, & \text{if } a_i = a_i',
0, & \text{Others}.
\end{array} \right.
\]
(17)

Define a weight vector \( w = [w_1, w_2, w_3, w_4, w_5]^T \), which satisfies \( w_j > \sum_{i=j+1}^5 w_i, 1 \leq j \leq 4 \), and the core of goal is detection as given follows.

**Definition 1:** The agent is considered to fully utilize the observations if and only if the agent chooses to adopt the observation when
\[
w^T \Phi(\sigma(q(k)), \sigma(\tilde{q}(k))) \geq w^T \Phi(\sigma(q(k)), \sigma(\bar{q}(k)))
\]
(18)

at time \( k \), and chooses the estimation otherwise.

If equation (18) holds, it means the agent aligns the actions with higher q-values. When comparing \( \Phi(\sigma(q(k)), \sigma(\tilde{q}(k))) \) and \( \Phi(\sigma(q(k)), \sigma(\bar{q}(k))) \), the first element of \( \Phi(\sigma(q(k)), \sigma(\tilde{q}(k))) \) being 1 indicates that the same action with maximum q value would be chosen based on \( q(k) \) and \( \tilde{q}(k) \). If the first element of \( \Phi(\sigma(q(k)), \sigma(\bar{q}(k))) \) is 0, then even if all remaining elements are 1, using \( \tilde{q}(k) \) is clearly superior to using \( \bar{q}(k) \). Mathematically, this judgment is satisfied if and only if the first element \( w_1 \) of the weight vector \( w \) is greater than the sum of all remaining elements, i.e., \( w_1 > \sum_{i=2}^5 w_i \). Only when the first element in both vectors is 1 do we proceed to compare whether the second elements of them are also 1, continuing this way until a difference is discerned. If \( w^T \Phi(\sigma(q(k)), \sigma(\tilde{q}(k))) > w^T \Phi(\sigma(q(k)), \sigma(\bar{q}(k))) \), discrepancies occur in the choices of more significant action with higher q-value based on \( q(k) \) and \( \tilde{q}(k) \). As a result, choosing estimation would lead to a negative reward, and vice versa. If \( \Phi(\sigma(q(k)), \sigma(\tilde{q}(k))) \) and \( \Phi(\sigma(q(k)), \sigma(\bar{q}(k))) \) equal, observation is more likely to be used, which will bring more information to the estimation of the next time step. Therefore, if an agent is considered to fully utilize the observations, it always chooses the closest result to the real action selection and also regarded as minimizing the impact of FDI attacks.

To encourage the agent to fully utilize the observations, the reward is defined as
\[
r(s(k), a(k)) = \begin{cases} 
R_{00}, & \text{if } \xi(k) \geq 0 \text{ and } a(k) = 0, \\
R_{10}, & \text{if } \xi(k) < 0 \text{ and } a(k) = 0, \\
R_{01}, & \text{if } \xi(k) \geq 0 \text{ and } a(k) = 1, \\
R_{11}, & \text{if } \xi(k) < 0 \text{ and } a(k) = 1,
\end{cases}
\]
(19)
where \( \xi(k) \) is defined as Table III. \( R_{00} \) and \( R_{11} \) are positive constants, and \( R_{10} \) and \( R_{01} \) are negative constants to be designed. In fact, all variables used to calculate \( \xi(k) \) are contained in \( s(k) \), so \( \xi \) is determined by \( s \) for any time. According to the definition of Table III, the following relations can be obtained: \( S^0 \cup S^1 = S \) and \( S^0 \cap S^1 = \emptyset \). If \( s(k) \in S^0 \), it implies that \( \xi(k) \geq 0 \). Conversely, if \( s(k) \in S^1 \), it indicates that \( \xi(k) < 0 \).

**Remark 4:** Since Assumption 1, Assumption 2, and the fixed parameters of the estimator network \( E_{o_2} \) and the controller network \( Q_{o_2} \), in the detection problem, the state transition probability \( P(s'(s, a)) \) is unknown but stationary and satisfies the Markov property.

2) **RL Solution and Feasibility Analysis:** The optimization objective of the detection is to learn an optimal policy \( \pi^*(s(k)) \) that maximizes the cumulative expected reward
\[
E \left[ \sum_{i=0}^{\infty} \gamma^i r(s(k+i), a(k+i)) \right], \quad \gamma \in [0, 1]
\]

\( \gamma \) is the discount factor. We employ a value-based RL algorithm to solve this problem.

For normal MDP problems, the optimal value function \( Q^*(s, a) = \max_{a'} Q^x(s, a) \) can be obtained by the following Q-learning iterative equation
\[
Q_{t+1}(s(k), a(k)) = Q_t(s(k), a(k)) + \eta_t \left[ r(s(k), a(k)) + \gamma \max_{a'} Q_t(s(k+1), a') - Q_t(s(k), a(k)) \right].
\]
(20)
where \( \eta_t \) is the learning rate and \( t \) denotes the number of iteration. If the exploration is sufficient and \( \eta_t \) is designed reasonably, \( Q_t(s, a) \) converges to \( Q^*(s, a) \) for any \( s \) and \( a \) with probability 1 when \( t \rightarrow \infty \) [38]. Then, the optimal policy \( \pi^* = \pi^*(s) = \argmax_s Q^*(s, a) \).

However, since the attacker is unobservable, the detection problem can only be modeled as a POMDP, and we can only find the optimal action-value \( Q^o(o, a) \). Similar to (20), it is
obtained the following iterative equation

\[ Q^0_{t+1}(o(k), a(k)) = Q^0_o(o(k), a(k)) + \eta t [r(s(k), a(k)) + \gamma \max_{a'} Q^0_o(o(k+1), a')] - Q^0_o(o(k), a(k)) \]

(21)

where \((o(k), a(k), r(s(k), a(k)), o(k+1))\) is the sampled data, and \(Q^0_o(o, a)\) is the action-value function based on observation \(o\) and action \(a\) in the \(t\)-th iteration. The convergence of (21) is more complicated since the existence of perceptual aliasing, which means different hidden states can result in the same observations, i.e., \(s \in S^0\) and \(s \in S^1\) may produce the same observation. To identify the constraints of perceptual aliasing on RL parameters, the following assumption and lemmas from [39] are adopted.

**Assumption 3:** [39] Every stationary policy makes the underlying MDPs in the POMDP ergodic.

**Lemma 1:** [39] In a POMDP satisfying Assumption 3, if a persistent excitation policy \(\mu\) is followed during learning, the iterative equation (21) will converge to the solution of the following system of equations with probability 1 (under the same conditions required for convergence in MDPs, plus the condition that the learning rates \(\eta_t\) is non-increasing):

\[ Q^* (o, a) = \sum_{s \in S} P^\mu(s|o)[r(s, a) + \gamma \sum_{a' \in A} P(o'|s, a) \max_{a'} Q^* (o', a')] \]

(22)

where \(P(o'|s, a) = \sum_{s' \in S} P(s'|s, a) P(o'|s')\).

Q-learning is a classic off-policy RL algorithm, which allows the strategy of collecting samples to be different from the final learned strategy. As a result, the utilization of persistent excitation policy \(\mu\) for sampling is easily satisfied. However, even when \(Q^* (o, a)\) is obtained, \(Q^0_o(o, a)\) and \(Q^* (s, a)\) may be different, and different reward setting may lead to different optimal solutions in the preference sense. For further analysis, the following lemma is proposed.

**Lemma 2:** If \(S \neq \emptyset\), the agent using \(Q^0_o(o, a)\) to choose actions cannot be considered to fully utilize the observations at all times.

**Remark 5:** Lemma 2 illustrates the limitations of detection algorithm due to the perceptual aliasing. However, in practical applications, perceptual aliasing occurs only in a small portion of the space. Furthermore, if perceptual aliasing occurs, it suggests that the system deviates only minimally from normal operation. Therefore, selecting either the observation or the estimation is a reasonable approach.

**Remark 6:** Similar to [13], it is assumed that the labeled training data is available, which means that during the training process, both the hidden states and the rewards are known and available. However, during the detection process, neither of them can be obtained.

Although Lemma 2 illustrates the limitation brought by perception, we can still set reasonable RL parameters including \(r(s, a)\) and \(\gamma\) to make sure that the agent fully utilizes the observations when \(s \in S\). When \(s \in \bar{S}\), different reward will lead to different behavior tendency of the agent. The conclusion is given by Theorem 1 and Theorem 2, and the relevant proof is given in the appendix.

**Theorem 1:** If the following conditions related to RL parameters hold,

1. \(R_{00} + R_{10} \geq 0\),
2. \(0 \leq \gamma < \gamma_1^* = \frac{R_{10} - 2R_{11} + R_{00}}{R_{10} - 2R_{11} + 3R_{00}} < 1\),
3. \(0 \leq \gamma < \gamma_2^* = \frac{R_{11} - R_{10}}{R_{11} - R_{10} + R_{00}}\)

then, the optimal action-value \(Q^* (s, a)\) encourages the agent to fully utilize the observations when \(s \in \bar{S}\), and to accept estimations when \(s \in \bar{S}\).

Theorem 1 provides a design rule for RL parameters that makes the agent more inclined towards the estimations when the perceptual aliasing occurs. Conversely, Theorem 2 provides a rule that makes the agent more inclined towards the observations.

**Theorem 2:** If the following conditions related to RL parameters hold,

1. \(R_{11} + R_{01} \geq 0\),
2. \(0 \leq \gamma < \gamma_1^* = \frac{R_{10} - 2R_{11} + R_{00}}{R_{10} - 2R_{11} + 3R_{00}} < 1\),
3. \(0 \leq \gamma < \gamma_2^* = \frac{R_{11} - R_{10}}{R_{11} - R_{10} + R_{00}}\)

then, the optimal action-value \(Q^* (s, a)\) encourages the agent to fully utilize the observations when \(s \in \bar{S}\), and to accept observations when \(s \in \bar{S}\).

Theorems 1 and 2 essentially achieve a balance between missed alarms and false alarms through the reward setting. In this paper, we consider that when perceptual aliasing occurs, it implies that even in the presence of an attack, the current state of the ego vehicle doesn’t deviate significantly. From the perspective of obtaining more information for the neural network, this paper adopts a reward design rules (26)-(28).

Such a detection scheme ensures that the optimal solution uses observations with minimal pollution under high-frequency attacks. However, the optimal strategy based on the existence of an attack [13] may cause the estimator to fail to obtain observations and subsequently diverge.

**3) DRL Solution:** Due to the complex and large-scale state space, we employ the classical DQN algorithm for training, utilizing a deep neural network \(Q_{\theta_1}\) with parameters \(\theta_1\) to approximate the Q-values of detection actions. The deep neural network adopts an encoder-decoder framework. The input \(o(k) = \{\bar{X}(k), \hat{X}(k), A(k)\}\) is a set, which includes three types of input information with different dimensions. Thus, different networks are used as encoders to extract input features, which are then concatenated to obtain the overall features of the input. Finally, two FCs serve as the decoder to obtain the Q-values for each action. See Fig. 3 for details.

To improve performance, tricks such as target network, experience replay and gradient clipping are also used. The
Fig. 3. The neural network architecture used in the detection.

following loss function is adopted.

$$L(\theta_3) = \sum_{b_3} \left( r(s, a) + \gamma \max_{a'} Q'(o', a'; \theta_3) - Q'(o, a; \theta_3) \right)^2,$$

where $Q'(o, a; \theta_3)$ is the target network. The deployment of the whole system is shown in Fig. 4, and the training process of the detection algorithm under the FDI attacks is summarized in Algorithm 2.

**Algorithm 2 Training Process of the Detection Algorithm**

**Under the FDI Attacks**

**input:** $\theta_1^0, \theta_2^0$ and trained safety layer.

**output:** $\theta_3^0$.

**initialization:** $\mathcal{B}'$: empty replay buffer

$\mathcal{X}$: training timesteps, $\theta_3, \theta_3'$: initial network parameters, $\mathcal{U}'$: update frequency.

$k = 0$, $k' = 0$.

while $k < \mathcal{X}'$ do

if $k' > N + 1$:

Calculate $\tilde{X}(k - 1)$;

Choose $a(k)$ according to $Q_{\theta_3}$ by $\epsilon$-policy;

Calculate $\tilde{q}(k) \tilde{q}(k)$ and $q(k)$ according to (14)-(16);

Calculate reward $r(s(k), a(k))$ according to (19);

Add $\{o(k), a(k), r(s(k), a(k)), o(k + 1)\}$ to $\mathcal{B}'$;

else:

Choose $a(k) = 0$;

end if

Select $a_{\pi}(k)$ according to (12);

Deploy $a_{\pi}(k)$ to the environment;

if the environment arrives a terminal:

reset the environment;

$k' = 0$;

end if

if $k > b_{33}$:

Sample $b_{33}$ tuples from $\mathcal{B}'$

Update $\theta_3$ according to (29);

if $kk' \equiv 0$

$\theta_3' \leftarrow \theta_3$;

end if

end if

V. EXPERIMENTS

We use the autonomous driving simulation environment highway-env v1.5 [40] to verify the algorithm designed in this paper. The three networks are trained respectively on a desktop computer equipped with an Intel i7-10700 CPU, 32 GB memory, and a Nvidia GeForce RTX 3060Ti GPU with 8 GB memory. In the simulation environment, the ego vehicle has minimum velocity $v_{\text{min}} = 17m/s$, and maximum velocity $v_{\text{max}} = 33m/s$. Each episode is 40s, and the control frequency of the high-level discrete actions is 1Hz. These settings refer to the environment’s default values and are consistent with studies [30], [32], which is based on the same environment. Scaling vector is set to $\mathcal{N} = [100, 100, 30, 30]^\top$. Other parameters follow the default conditions of the environment. Since the rewards do not intuitively reflect the performance of a controller, and considering the reward setting (7), we use longitudinal average speed and safety rate as evaluation metrics, which are commonly used to assess efficient and safe driving behavior [32], [41], [42], [43], [44]. The longitudinal average speed (LAS) refers to the speed of the ego vehicle in the forward direction. Compared with the absolute speed, this metric can better reflect the traffic efficiency.

In the following experiments, we carry out 10 trials with different seeds. Each trial runs for 100 episodes, and we then calculate the mean and standard deviation of both LAS and safety rate. The safety rate is calculated based on the proportion of non-collision instances in 100 episodes, and also indicates the collision probability of the ego vehicle.

We first independently validate the effectiveness of the estimation model. The hyperparameters of the estimator network are shown in Table IV, where ConvLSTM $i$ and FC $j$ indicates the $i$-th ConvLSTM layer and $j$-th linear layer of the corresponding component, respectively. The maximum of $i$ and $j$ listed in the table is the number of layers of the corresponding network.

The elements of the weight matrix $W$ are designed as

$$w_{ahl} = \begin{cases} 1, & b \leq h_1, \\ 1 - w_h \frac{b - h_1}{h_2 - h_1}, & \text{Others,} \end{cases}$$

where $h_1$ is the threshold that distinguishes the importance of surrounding vehicles, which is set to 4 in this scene. The parameter $w_h \in [0, 1]$ is a coefficient. $w_h = 1$ means the furthest vehicles are completely disregarded. When $w_h = 0$, all the surrounding vehicles are considered to be equally important. In this article, we set $w_h = 0.5$. The network framework of [22] is introduced as a baseline for the learning-based scheme of the estimation algorithm, and the relevant structure and parameters are modified to make them consistent with the proposed network. For newly discovered targets, we simply pad their historical trajectories with zeros.

Fig. 4. Deployment of the whole system.
The performance of the proposed algorithm under FDI attacks are shown as follows. We assume that the attack signal has a frequency of $P(\alpha) = \rho$, and its amplitude follows a uniform distribution with a mean of 0 and a standard deviation equal to $e$ times the amplitude of the original signal.

The parameters of the detector including network and DRL are shown in Table VII.

First, we compare the difference between the reward setting according to Theorem 1 and Theorem 2. We set $\gamma = 0.2$, $R_{00} = 1$, $R_{10} = -0.5$, $R_{01} = -0.2$, and $R_{11} = 0.2$ to satisfy Theorem 2 as parameter I. Then, the parameter II we test is $\gamma = 0.2$, $R_{11} = 1$, $R_{01} = -0.5$, $R_{10} = -0.2$, and $R_{00} = 0.2$ to satisfy the Theorem 1. The experimental results are shown in Table VIII.

We use a high-frequency small attack ($p = 0.5$ and $e = 0.1$) and a low-frequency large attack ($p = 0.1$ and $e = 1$) for simulation. From Table VIII, it is concluded that the parameter I is superior to the parameter II, particularly when the attack frequency is high and the amplitude is small, which means more perceptual aliasing. In such situations, encouraging the acceptance of observations can bring more perceptual information to estimation in the next time, and lead to a better result. The small effect of perceptual aliasing is offset by the generalization ability of the ESA-DQN with safety layer. Even with a lower attack frequency, setting according to Theorem 2 also yields a good result.

The model is also verified in the testing environment, which is assumed to have a delay of less than 1s in the observations.

The proposed estimation algorithm performs better than the model-based method. A possible reason is that the model-based method fail to perceive the maneuver of the target vehicle, leading to a higher failure rate in scenarios with frequent lane changes. Therefore, the safety rate has a lower mean and a larger standard deviation. Compared to the existing learning-based estimation algorithm, the proposed method does not show significant improvement in RMSE within $[0, \mathcal{H}]$. However, it provides more accurate estimates of vehicles that are closer to the ego vehicle, thereby improving safety rates and reducing variance.

During the training process, the dataset is divided into training and validation sets in a ratio of 6:1. Their estimation performance is first evaluated as Table V using the root of the mean squared error (RMSE) as the metric. Starting from the ego vehicle, evaluate the estimate accuracy of surrounding vehicles within the ranges $[0, h_1]$, $[h_1 + 1, \mathcal{H}]$, and $[0, \mathcal{H}]$.

The model is also verified in the testing environment, which is assumed to have a delay of less than 1s in the observations of the ego vehicle. It means we always use $\hat{X}(k)$ as the input of the controller network $Q_{nt}^\ast$ at time $k$, and the uncompromised history state $H(k)$ is always available for obtaining an estimate of $\hat{X}(k + 1)$.

The comparison result is shown in the Table VI.

The proposed estimation algorithm performs better than the model-based method. A possible reason is that the model-based method fail to perceive the maneuver of the target vehicle, leading to a higher failure rate in scenarios with frequent lane changes. Therefore, the safety rate has a lower mean and a larger standard deviation. Compared to the existing learning-based estimation algorithm, the proposed method does not show significant improvement in RMSE within $[0, \mathcal{H}]$. However, it provides more accurate estimates of vehicles that are closer to the ego vehicle, thereby improving safety rates and reducing variance.

The performance of the proposed algorithm under FDI attacks are shown as follows. We assume that the attack signal has a frequency of $P(\alpha) = \rho$, and its amplitude follows a uniform distribution with a mean of 0 and a standard deviation equal to $e$ times the amplitude of the original signal.

The parameters of the detector including network and DRL are shown in Table VII.

First, we compare the difference between the reward setting according to Theorem 1 and Theorem 2. We set $\gamma = 0.2$, $R_{00} = 1$, $R_{10} = -0.5$, $R_{01} = -0.2$, and $R_{11} = 0.2$ to satisfy Theorem 2 as parameter I. Then, the parameter II we test is $\gamma = 0.2$, $R_{11} = 1$, $R_{01} = -0.5$, $R_{10} = -0.2$, and $R_{00} = 0.2$ to satisfy the Theorem 1. The experimental results are shown in Table VIII.

We use a high-frequency small attack ($p = 0.5$ and $e = 0.1$) and a low-frequency large attack ($p = 0.1$ and $e = 1$) for simulation. From Table VIII, it is concluded that the parameter I is superior to the parameter II, particularly when the attack frequency is high and the amplitude is small, which means more perceptual aliasing. In such situations, encouraging the acceptance of observations can bring more perceptual information to estimation in the next time, and lead to a better result. The small effect of perceptual aliasing is offset by the generalization ability of the ESA-DQN with safety layer. Even with a lower attack frequency, setting according to Theorem 2 also yields a good result.

Next, we select the latest DRL-based detection research [13] as the baseline. We experimented with multiple sets of attack parameters, including different attack frequencies ($p \in [0.1, 0.2, 0.3, 0.5]$) and magnitudes ($e \in [0.1, 0.2, 0.5, 0.7, 1]$). For each set of parameters, we conducted repeated experiments and calculated the standard deviation of the mean values. The comparative results are depicted in Fig. 5 and Fig. 6.

From Fig 5 and Fig. 6, it can be observed that, with approximately the same LAS, our algorithm significantly improves

<table>
<thead>
<tr>
<th>Component</th>
<th>Item</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information gridization</td>
<td>$l$</td>
<td>20</td>
</tr>
<tr>
<td>ConvLSTM 1</td>
<td>Convolutional kernel</td>
<td>$3 \times 3$</td>
</tr>
<tr>
<td>ConvLSTM 2</td>
<td>Convolutional kernel</td>
<td>$3 \times 3$</td>
</tr>
<tr>
<td>SE block</td>
<td>Reduction</td>
<td>16</td>
</tr>
<tr>
<td>FC 1</td>
<td>Number of neurons</td>
<td>512</td>
</tr>
<tr>
<td>FC 2</td>
<td>Number of neurons</td>
<td>128</td>
</tr>
<tr>
<td>Others</td>
<td>$N$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Weight decay</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>$b_{tr}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Epochs</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RMSE $[0, h_1]$</th>
<th>RMSE $[h_1 + 1, \mathcal{H}]$</th>
<th>RMSE $[0, \mathcal{H}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>0.3135</td>
<td>0.5347</td>
<td>0.4243</td>
</tr>
<tr>
<td>Learning-based [22]</td>
<td>0.4142</td>
<td>0.4391</td>
<td>0.4280</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LAS (m/s)</th>
<th>Safety rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>$30.42 \pm 0.51$</td>
<td>$0.98 \pm 0.02$</td>
</tr>
<tr>
<td>Model-based [12]</td>
<td>$30.53 \pm 1.06$</td>
<td>$0.90 \pm 0.05$</td>
</tr>
<tr>
<td>Learning-based [22]</td>
<td>$30.41 \pm 0.88$</td>
<td>$0.96 \pm 0.05$</td>
</tr>
</tbody>
</table>
the safety rate compared to the baseline, especially at higher attack frequencies. When the attack amplitudes are high, the optimal solutions for both design approaches tend to be consistent, resulting in similar performance. However, at higher attack frequencies, our method can effectively leverage the observation information to improve the safety rate.

VI. CONCLUSION

This paper has proposed a secure and safe control strategy of CAVs. Specifically, this strategy divides the system design into two sub-problems. In the estimation sub-problem, the state estimations of all the surrounding vehicles are treated as an entity. An estimation algorithm combining deep learning and a physical model has been proposed. In the control sub-problem, an ESA-DQN controller with safety layer has been employed. Diverging from the objectives of most existing schemes in the FDI attack detection problem, the detector has been designed to guide the ego vehicle to fully utilize the observations, and to minimize the impact of FDI attacks. Different reward settings leading to various decision biases have been theoretically analyzed. The DQN algorithm has been employed to obtain an approximate solution for the detector. Finally, the effectiveness of the algorithm has been verified by several simulation experiments. Our future efforts will focus on reshaping reward function of the safe controller to cover more comprehensive objectives and extending this work to multi-agent environments.

APPENDIX

Proof: [Proof of Lemma 2] Suppose that $s(k_1) \in \mathcal{S}^0$ and $s(k_2) \in \mathcal{S}^1$ are associated with the observations $o(k_1)$ and $o(k_2)$, respectively. In addition, $o(k_1) = o(k_2)$. Since the function $Q^\pi(o, a)$ produces the same output for the same input, argmax$_a Q^\pi(o(k_1), a) = \text{argmax}_a Q^\pi(o(k_2), a)$. However, if the agent is considered to fully utilize the observations, then argmax$_a Q^\pi(o(k_1), a)$ should be equal to 0 and argmax$_a Q^\pi(o(k_2), a)$ should be equal to 1. Thus, the agent cannot be considered to fully utilize the observations in the state of perceptual aliasing. This completes the proof.

Proof: [Proof of Theorem 1] Condition (24) implies $R_{01} - 2R_{00} + R_{11} > 0$, which means $R_{11} > R_{00}$ and $R_{01} + R_{11} > 0$. Next, we first define the function

$$f_1(\gamma) = \gamma R_{11} + (R_{00} - R_{01})(1 - \gamma),$$

whose domain is $[0, 1]$. It is obvious that $f_1(\gamma_1^*) = 0.5$. Next, we analyze the monotonicity of $f_1(\gamma)$.

$$\frac{\partial f_1(\gamma)}{\partial \gamma} = \frac{\gamma R_{11} - (R_{00} - R_{01})(\gamma - 1)}{(R_{11} - R_{01})(\gamma - 1)^2}
- \frac{R_{01} - R_{00} + R_{11}}{(R_{11} - R_{01})(\gamma - 1)^2} > 0 \quad (32)$$

Since $f_1(\gamma)$ is a continuous function in $[0, 1]$, $f_1(\gamma) < 0.5$ always holds, when $\gamma < \gamma_1^*$.

Due to the perceptual aliasing, the hidden state recognition rate $\rho$ in this problem is defined as

$$\rho = \min_{o \in \mathcal{O}} \max\{\tilde{\rho}(o), 1 - \tilde{\rho}(o)\}, \quad (33)$$

where $\tilde{\rho}(o) = \sum o \in \mathcal{O} P(o)\tilde{P}(o)$. According to the form of (33), we have $\rho \geq \frac{1}{2} > \frac{R_{11} + (R_{01} - R_{00})\gamma}{(1 - \gamma)(R_{11} - R_{01})},$ then

$$\gamma R_{11} \frac{1}{1 - \gamma} < \rho(R_{11} - R_{01}) - (R_{00} - R_{01}). \quad (34)$$

For any $o \in \mathcal{O}$, $P(o'|o)r(s, a) \leq r(s, a) \leq R_{11}$. According to (22) and reward setting in (19), we have

$$Q^\pi(o, a) \leq R_{11} + \gamma P^\pi(o|s) \sum o' \in \mathcal{O} P(o'|s, a) \max_{a'} Q^\pi(o', a') \leq \sum_{i=0}^{\infty} (\gamma R_{11})^i = \frac{R_{11}}{1 - \gamma}. \quad (35)$$

On the other hand,

$$Q^\pi(o, a) \geq \min(\rho R_{11} + (1 - \rho)R_{01}, \rho R_{00} + (1 - \rho)R_{10}) + \gamma P^\pi(o|s) \sum o' \in \mathcal{O} P(o'|s, a) \max_{a'} Q^\pi(o', a'). \quad (36)$$

Based on (23), note that $\rho \geq \frac{1}{2}$, we have $Q^\pi(o, a) \geq 0$.

For convenience, we divide the observation space $\mathcal{O}$ into two subspaces $\mathcal{O}^0 \in \mathcal{O}$ and $\mathcal{O}^1 \in \mathcal{O}$, which satisfies $\mathcal{O}^0 \cup \mathcal{O}^1 = \mathcal{O}$ and $\mathcal{O}^0 \cap \mathcal{O}^1 = \emptyset$. If $o \in \mathcal{O}^1$, then $\tilde{\rho}(o) \geq 0.5$, which means this $o$ is more likely to be an observation of the state suffering from FDI attacks. If $o \in \mathcal{O}^0$, then $\tilde{\rho}(o) < 0.5$, whose mathematical meaning is the opposite of the former.

Next, we need to prove that $Q^\pi(o, 1) > Q^\pi(o, 0)$, if $o \in \mathcal{O}^1$, we have

$$Q^\pi(o, 1) \geq \rho R_{11} + (1 - \rho)R_{01} + \gamma \min_{o, a} Q^\pi(o, a), \quad (37)$$

Authorized licensed use limited to: Southeast University. Downloaded on March 22,2024 at 02:20:12 UTC from IEEE Xplore. Restrictions apply.
and \[Q^{o^*}(o, 0) \leq R_{00} + \frac{\gamma R_{11}}{1 - \gamma}.\] (38)

Based on (34), subtracting (38) from (37) we obtain
\[
Q^{o^*}(o, 1) - Q^{o^*}(o, 0) \geq \rho R_{11} + (1 - \rho)R_{01} - \frac{\gamma R_{11}}{1 - \gamma} > 0.
\]

Thus, \(Q^{o^*}(o, 1) > Q^{o^*}(o, 0)\). For \(s \in \tilde{S}\), it is equivalent to substituting \(\rho = 1\) in equation (39). Since equation (39) always holds for \(\rho \geq 0.5\), it also holds for \(\rho = 1\). As for \(s \notin \tilde{S}\), \(Q^{o^*}(o, 0) \geq 0\), and \(Q^{o^*}(o, 1) \leq 0 + \frac{\gamma R_{11}}{1 - \gamma}\). The same as (39), we have
\[
Q^{o^*}(o, 1) - Q^{o^*}(o, 0) \geq (R_{00} - R_{01}) - \frac{\gamma R_{11}}{1 - \gamma} \geq 0.
\]

Define \(\bar{f}(\gamma) = (1 - \gamma)(R_{00} - R_{01}) - \gamma R_{11}, \) we have \(\bar{f}(0) = R_{00} - R_{01} > 0\). Thus, \(\bar{f}(\gamma) > 0\) always holds when \(\gamma < \tilde{\gamma}_2\). Therefore,
\[
Q^{o^*}(o, 0) - Q^{o^*}(o, 1) > 0.
\]

(39) and (41) indicate that \(Q^{o^*}\) encourages the detection agent to fully utilize the observations in regions without perceptual aliasing. This completes the proof.

**Proof:** [Proof of Theorem 2] Condition (27) implies \(R_{10} - 2R_{11} + R_{00} > 0\), which means \(R_{00} > R_{11}\) and \(R_{10} + R_{00} > 0\). Next, we first define the function \(f(\gamma) = \gamma R_{00} + (R_{11} - R_{10})(1 - \gamma),\)
\[
\text{whom whose domain is } [0, 1]. \text{ It is obvious that } \bar{f}(\tilde{\gamma}_2^*) = 0.5.
\]

Since \(\bar{f}(\gamma)\) is a continuous function and \(\bar{f}(\gamma) > 0\) in \([0, 1], \ bar{f}(\gamma) < 0.5\) always holds, when \(\gamma < \tilde{\gamma}_2^*\). Thus, \(\gamma R_{00} / (1 - \gamma) < R_{00} - R_{11} + (1 - \gamma)R_{10}\).
\[\frac{\gamma R_{00}}{1 - \gamma} < \rho (R_{10} - R_{11}).\]

Similar to the proof of Theorem 2, the case where \(o \in \Phi^o\) is considered first. If \(o \in \Phi^o\), we have
\[
Q^{o^*}(o, 0) \geq \rho R_{00} + (1 - \rho)R_{10} + \gamma \min_{a, a'} Q^{o^*}(o, a'),\]
and
\[
Q^{o^*}(o, 1) \leq R_{11} + \frac{\gamma R_{00}}{1 - \gamma}.
\]

Based on (43), subtracting (45) from (44) we obtain
\[
Q^{o^*}(o, 0) - Q^{o^*}(o, 1) \geq \rho R_{00} + (1 - \rho)R_{10} - \frac{\gamma R_{00}}{1 - \gamma} > 0.
\]

For \(s \in \tilde{S}\), \(Q^{o^*}\) encourages the agent to choose observations. When \(o \in \Omega^1\) and \(s \in \tilde{S}\), we have \(Q^{o^*}(o, 1) \geq R_{11},\) and \(Q^{o^*}(o, 0) \leq R_{10} + \frac{\gamma R_{00}}{1 - \gamma}\). The same as (46), we have
\[
Q^{o^*}(o, 1) - Q^{o^*}(o, 0) \geq (R_{11} - R_{10}) - \frac{\gamma R_{00}}{1 - \gamma} > 0.
\]

Define \(\bar{f}'(\gamma) = (1 - \gamma)(R_{11} - R_{10}) - \gamma R_{00}\), we have \(\frac{\partial \bar{f}'(\gamma)}{\partial \gamma} = (R_{11} - R_{10}) - R_{00} < 0\), and \(\bar{f}'(0) = R_{11} - R_{10} > 0\). Thus, \(\bar{f}'(\gamma) > 0\) always holds, if \(\gamma < \tilde{\gamma}_2^*\). Therefore,
\[
Q^{o^*}(o, 1) - Q^{o^*}(o, 0) > 0.
\]

(46) and (48) imply that \(Q^{o^*}\) encourage the detection agent to fully utilize the observations in regions without perceptual aliasing. This completes the proof.

**REFERENCES**


Guoxi Chen (Student Member, IEEE) received the B.E. degree in automation from Southeast University in 2021, where he is currently pursuing the Ph.D. degree in control engineering. His major research interests include autonomous driving, reinforcement learning, and network security.

Tiejun Wu received the B.S. degree in computer science and technology and the master’s degree in software engineering from Huazhong University of Science and Technology in 2002 and 2006, respectively. He is currently pursuing the Ph.D. degree with Southeast University. His research interests include cyber security and traffic analysis.

Xinde Li (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Department of Control Science and Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2007. Afterward, he joined the School of Automation, Southeast University, Nanjing, China, where he is currently a Professor and the Ph.D. Supervisor. His research interests include information fusion, object recognition, computer vision, and intelligent robot.

Ya Zhang (Senior Member, IEEE) received the B.S. degree in applied mathematics from China University of Mining and Technology in 2004 and the Ph.D. degree in control engineering from Southeast University in 2010. Since 2010, she has been with Southeast University, where she is currently a Professor with the School of Automation. Her research interests include multigain systems, reinforcement learning, and network security.